Design of Antenna Arrays Using Chaotic Jaya Algorithm

R. Jaya Lakshmi and T. Mary Neebha

Abstract Antenna arrays are commonly used in communication systems to enhance desired radiation properties. The optimum design of antenna arrays is an important problem as it greatly affects the efficiency of the communication system. In this work, a variation of Jaya algorithm, i.e., Chaotic Jaya, is used to design planar antenna arrays to get minimized side lobe levels (SLL) thus improving the directivity. Two design examples of planar antenna arrays (concentric circular and hexagonal) are formulated by considering different inter-element spacings. Comparison of results between Chaotic Jaya algorithm and few other algorithms is done to demonstrate the algorithm's efficiency. The Chaotic Jaya algorithm was able to get much-reduced side lobe levels.

Keywords Optimization · Planar antenna array synthesis · Chaotic Jaya algorithm

1 Introduction

Antenna arrays find widespread applications in communications such as mobile, television transmissions in UHF and VHF frequency ranges, satellite, sonar, radar, etc. Antenna arrays are used commonly in satellite, mobile and radar communications systems to have improved performance in terms of directivity, signal-to-noise ratio (SNR), spectrum efficiency, etc. Therefore, the proper design of antenna arrays is a crucial factor in deciding the system's performance. Linear antenna arrays do not give efficient performance in all azimuth directions, despite having high directivity, whereas a circular array can perform all azimuth scan around its center. Hence,

R. Jaya Lakshmi (✉)
Department of Information Engineering and Computer Science, University of Trento, 381243 Trento, Italy
e-mail: jayalakshmi.ravipudi@studenti.unitn.it

T. Mary Neebha
Karunya Institute of Technology and Sciences, Coimbatore, India
e-mail: maryneebha@karunya.edu

R. Venkata Rao and J. Taler (eds.), *Advanced Engineering Optimization Through Intelligent Techniques*, Advances in Intelligent Systems and Computing 949,
https://doi.org/10.1007/978-981-13-8196-6_30

concentric circular antenna arrays (CCA) find applications in direction of arrival (DOA). Uniform circular arrays (UCCA) are the ones which have inter-element spacing of about half a wavelength.

An antenna array's radiation pattern is dependent on the array structure, number of ON and OFF elements, inter-element spacing and individual elements' amplitude and phase excitation, etc. One method of suppressing side lobe levels is array thinning. Thinning method for linear arrays is reported in the literature [7, 8] and can be very well used for circular and hexagonal arrays. Thinning means, at a time, only few elements of the array are ON condition while others are OFF. ON means the element is provided with feed, and OFF means either the element is not provided with feed or is connected to a dummy or matched load. This helps to reduce cost by decreasing the number of elements required and power consumption.

Design of arrays with large number of elements using thinning is a difficult task as solving the functions require efficient techniques. In recent years, researchers have applied various algorithms to get efficient design of antenna arrays, such as enhanced particle swarm optimization (PSO) [9], modified PSO [10], ant colony optimization (ACO), genetic algorithm (GA), differential evolution (DE), flower pollination algorithm and cuckoo optimization algorithm [13]. For concentric circular arrays, algorithms such as PSO, improved particle swarm optimization (IPSO), gravitational search algorithm and modified PSO [4], comprehensive learning PSO [5], decomposition-based evolutionary multi-objective optimization approach [3] and backtracking search algorithm [6] have been used. However, the above-mentioned optimization algorithms have parameters that are specific to them, i.e., those parameters are also required to be tuned apart from the common parameters. Keeping this in view, Rao [11] proposed Jaya algorithm that excludes the effort of tuning any specific parameters of the algorithm and requires the user to just tune the common parameters such as size of population and no. of generations. The algorithm is simple in implementation along with giving efficient results. The detailed working and the variants of Jaya algorithm are discussed in Rao [12].

The configuration of the concentric circular and hexagonal antenna arrays is presented in Sect. 2. A description of Jaya and Chaotic Jaya algorithm is given in Sect. 3. Section 4 defines the objective function used, and Sect. 5 includes the synthesis of concentric circular and hexagonal antenna array. Conclusions are given in Sect. 6.

2 Design Equations

2.1 Concentric Circular Antenna Array

The configuration of uniform two-ring concentric circular array with *Np* elements along the circle with radius r_p ($p = 1, 2$) is given in Fig. 1 [1]. Inter-element spacing is d_p which is half the operational wavelength. The radius of *p*th circle is given by

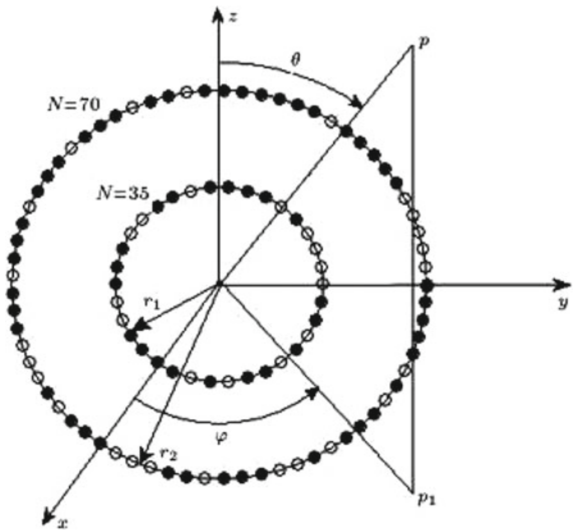

Fig. 1 Configuration of uniform two-ring concentric circular array antenna

$$r_p = N_p \lambda / 4\pi \tag{1}$$

The angular position is given by:

$$\omega_{pn} = 2n\pi / N_p \tag{2}$$

The array factor (AF) is calculated as follows.

$$\mathrm{AF}(\theta, \varphi) = \sum_{p=1}^{2} \sum_{n=1}^{N_p} I_{pn} e^{jkr_p\left[\sin\theta\cos(\emptyset-\emptyset_{pn})-\sin\theta_0\cos(\emptyset_0-\emptyset_{pn})\right]} \tag{3}$$

I_{pn} Excitation of nth element in pth ring
θ elevation angle
φ azimuthal angle
k $2\pi/\lambda$, wave number
$\theta_{0,0}$ direction of main beam's maximum.

$\theta_0 = 0$ and $\varphi_0 = 0$ are taken for the design problem and $= 0$ plane is considered. When the above-mentioned angles are substituted in Eq. (3), it gets reduced to:

$$\mathrm{AF}(\theta, \varphi) = \sum_{p=1}^{2} \sum_{n=1}^{N_p} I_{pn} e^{\left[jkr_p\sin\theta\cos(\emptyset-\emptyset_{pn})\right]} \tag{4}$$

2.2 Hexagonal Antenna Array

The configuration of uniform hexagonal antenna array is shown in Fig. 2. To visualize a hexagonal array, two concentric circles of radii r_1 and r_2 are considered. Each circle is having elements on it. The array consists of $2N$ elements: N on the mid-point of the sides of the hexagon and N on the vertices.

The array factor (AF) for this configuration is computed as given below:

$$\mathrm{AF}(\theta, \varphi) = \sum_{n=1}^{N} A_n \mathrm{e}^{jkr_1 \sin\theta(\cos\varphi_{1n}\cos\varphi + \sin\varphi_{1n}\sin\varphi)} + B_n \mathrm{e}^{jkr_2 \sin\theta(\cos\varphi_{2n}\cos\varphi + \sin\varphi_{2n}\sin\varphi)} \quad (5)$$

$$\varphi_{1n} = 2\pi(n-1)/N \quad (6)$$

$$\varphi_{2n} = \varphi_{1n} + \pi/N \quad (7)$$

$$r_1 = d_e/\sin\left(\frac{\pi}{N}\right) \quad (8)$$

$$r_2 = r_1\cos\left(\frac{\pi}{N}\right) \quad (9)$$

where A_n and B_n are relative amplitudes of nth element lying on the vertices and the mid-point, φ_{1n} is the angle between the nth element at the vertex point in XY plane and the x-axis, φ_{2n} is the angle between the nth element lying on the mid-point of the hexagon's side and x-axis, r_1 and r_2 are the radii of the two circles, and θ is the elevation angle, and φ is the azimuth angle.

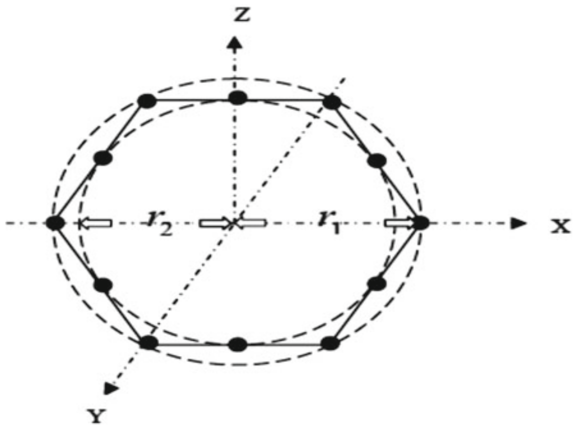

Fig. 2 Configuration of uniform hexagonal array antenna

3 The Jaya Algorithm

The first step in the implementation of Jaya algorithm is to generate initial solutions. Therefore, P initial solutions are generated in random fashion by keeping in consideration the design variables' range values, i.e., upper and lower bounds. Then, every solution's variable values are updated using Eq. (10). Let the objective function be denoted with f. For demonstration, it is considered that the function is to be minimized (or maximized, according to the design problem). Say, the no. of design variables is 'nd.' best_f is used to denote the best solution and worst_f to denote the worst solution [12].

$$\begin{aligned} A(x+1, y, z) = A(x, y, z) &+ r(x, y, 1)(A(x, y, b) - |A(x, y, z)|) \\ &- r(x, y, 2)(A(x, y, w) - |A(x, y, z)|) \end{aligned} \tag{10}$$

Among the population, the best and worst solutions are identified by their indices b and w. The indices to denote iteration, variable and the candidate solution x, y, z. $A(x, y, z)$ mean the yth variable of zth candidate solution in xth iteration. $r(x, y, 1)$ and $r(x, y, 2)$ are the randomly generated numbers in the range of [0, 1] and act as scaling factors and to ensure good diversification. Figure 3 depicts Jaya algorithm's flowchart.

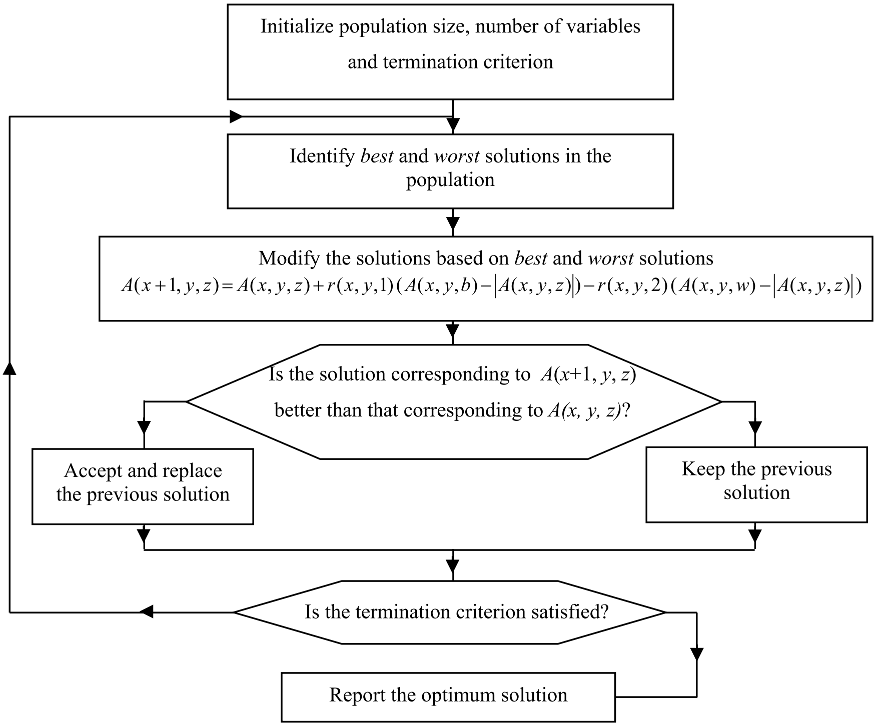

Fig. 3 Flowchart of Jaya algorithm

In this algorithm, the value of the objective function of every candidate solution of the population is improved. Values of variables of each solution are updated by the algorithm in order to move the corresponding function value closer to the best solution. After updating variable values, comparison between the new function value and corresponding old function value. Then, only the solutions having better objective function value (lower values for minimization and higher values for maximization problems) are included in the next generation.

3.1 Chaotic Jaya Algorithm

A variation of Jaya algorithm, i.e., Chaotic Jaya, is used in this paper. This variant helps in improving the speed of convergence and also improves the search space exploration without getting entrapped in the local optima. Chaos is a branch of mathematics that deals with systems that are sensitive to initial conditions. In chaos optimization algorithms, the chaotic maps are used to map the chaotic sequences to produce the design variables, instead of producing sequences that are simply random.

Different chaotic maps are available to introduce chaos in optimization, e.g., exponential map, circle map, logistic map and tent map. These maps are used to initialize population that is uniformly distributed so that the quality of initial population is improved. In these chaotic maps, an initial value is chosen in the range [0, 1] (or according to the range of chaotic map).

The only difference in the implementation Jaya algorithm and Chaotic Jaya algorithm is in the use of a chaotic map by the latter one. Chaotic Jaya uses the tent map chaotic function to generate random numbers, and the function is expressed by Eq. (11) [14].

$$x_{k+1} = \left\{ \begin{array}{ll} \frac{x_k}{0.7} & x_k < 0.7 \\ \frac{10}{3}(1 - x_k) & x_k \geq 0.7 \end{array} \right\} \tag{11}$$

where x_k is the previously generated chaotic random number and x_{k+1} is the newly generated chaotic random number.

4 Objective Function

The side lobe level is calculated by the following equation.

$$\mathrm{SLL} = 20\log_{10}\left(\frac{|\mathrm{AF}(\theta, \varphi)|}{|\mathrm{AF}(\theta, \varphi)|\mathrm{max}}\right) \tag{12}$$

For two-ring concentric circular antenna array, the fitness function is given as follows [1].

$$\begin{aligned} \text{Fitness} &= \max(\text{SLL}), \quad \text{FNBW}_o < \text{FNBW}_d \\ &= 10^2, \quad \text{otherwise} \end{aligned} \tag{13}$$

where FNBW is the first null beam width. The subscript o denotes the obtained FNBW and d denotes the desired FNBW. The desired FNBW for this work is 10°.

For the case of hexagonal antenna array, the objective function is formulated as follows [2].

$$\text{Fitness} = \min(\max(20\log|\text{AF}(\theta)|)) \tag{14}$$

$$\theta \in \left\{\left[-90^\circ, -|\text{FN}|^\circ\right] \text{and} \left[|\text{FN}|^\circ, 90^\circ\right]\right\}$$

where FN is first null in the antenna radiation pattern and $\text{AF}(\theta, \varphi)$ is given by Eq. (5). In all the cases, % thinning is calculated as the percentage of no. of OFF elements to total number of elements.

5 Case Studies

5.1 Synthesis of Concentric Circular Antenna Array

The inter-element spacing is considered as 0.5. Equation (4) is used to calculate the array factor, and Eq. (13) is the fitness function. This problem was worked out by [1] using firefly algorithm (FFA). In their paper, taking 50 as the population size and 500 as the maximum number of generation produced an optimal result for FFA algorithm. Here, the same case study is solved using Chaotic Jaya algorithm. To compare both algorithms fairly, the maximum number of function evaluations is kept same as that of FFA (i.e., 25,000). So population size is maintained as 50 and 500 is chosen as the maximum number of generations.

Table 1 shows the results of Chaotic Jaya algorithm for two-ring concentric circular array. By comparison, it is seen that minimum peak SLL by FFA is −18.36 dB whereas Chaotic Jaya algorithm resulted in a minimum peak SLL of −21.495287 dB which is an improvement over FFA results. Table 2 reports the amplitude distribution. The radiation pattern is shown in Fig. 4.

Table 1 Results of concentric circular

Algorithm	No. of ON elements	% Thinning	Max. SLL (dB)
Chaotic Jaya	53	49.52	−21.495287
[a]FFA	69	34.29	−18.36

[a]The results of FFA are reproduced from Basu and Mahanti [1]

Table 2 Amplitude distribution results of concentric circular array

Algorithm	Distribution of ON-OFF elements	
	Ring 1	Ring 2
Chaotic Jaya	0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 1 0 1 0 1 0 1 1 1 1 0 1 1 1 0 0 1 0	0 0 0 0 0 1 0 0 1 1 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 1 1 0 1 0 1 0 1 1 1 1 1 1 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1
[a]FFA	0 1 0 1 1 1 1 1 1 1 1 0 1 0 1 1 0 0 0 0 0 0 1 1 0 1 1 0 1 1 1 1 0 1 0	1 1 0 1 0 1 0 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 0 0 1 0 1 1 0 1 0 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 0 1 0 0 1

[a]The results of elements distribution using FFA are reproduced from Basu and Mahanti [1]

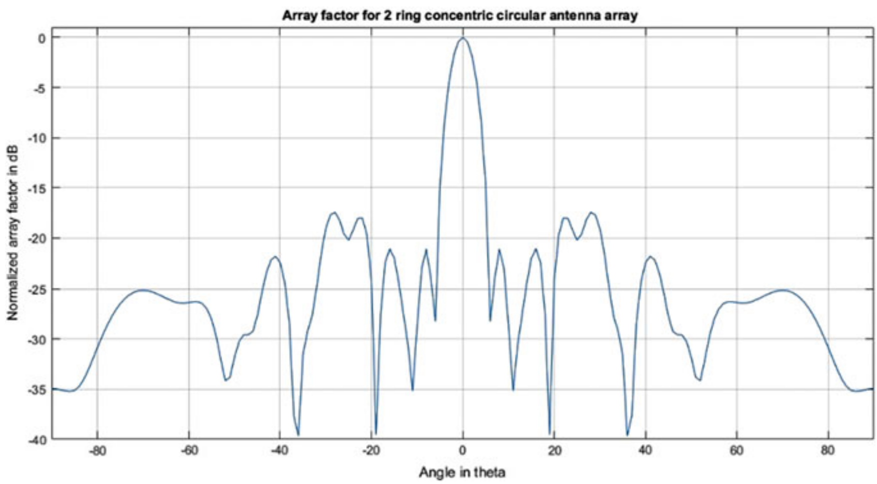

Fig. 4 Normalized array pattern of CCA

5.2 *Synthesis of Hexagonal Antenna Array*

For all the cases of hexagonal antenna array, Eq. (5) is used to calculate the array factor and Eq. (14) is the fitness function. The number of elements considered is $2N = 12$, i.e., six elements lie on the inner circle and the other six on the outer circle. This problem was worked out by [2] using improved particle swarm optimization (IPSO). In their paper, taking 50 as the population size and of 50 and 100 as the maximum number of generations gave the optimal result for IPSO algorithm. Here, the same case study is solved using Chaotic Jaya algorithm. To compare both algorithms fairly, the maximum number of function evaluations is kept same as that of IPSO (i.e., 5000). So population size is maintained as 50 and 100 is chosen as the maximum number of generations.

Case 1: 0.50λ spacing

The inter-element spacing is considered as 0.5λ. The optimized array patterns obtained by Chaotic Jaya algorithm for reducing SLL for 12-element array and of IPSO [2] is tabulated in Table 3. From Table 3, it is perceived that minimum peak SLL by IPSO is −23.85 dB whereas Chaotic Jaya algorithm resulted in a minimum peak SLL of −27.605378 which is an enhancement of IPSO results. The radiation pattern by Chaotic Jaya algorithm along with the pattern of a fully populated array is plotted in Fig. 5. Table 4 reports the amplitude distribution.

Case 2: 0.55λ spacing

Table 3 Results of hexagonal antenna array (0.50λ)

Algorithm	No. of ON elements	% Thinning	Max. SLL (dB)
Chaotic Jaya	6	50	−27.605378
[a]IPSO	6	50	−23.85

[a]The results of IPSO are reproduced from Bera et al. [2]

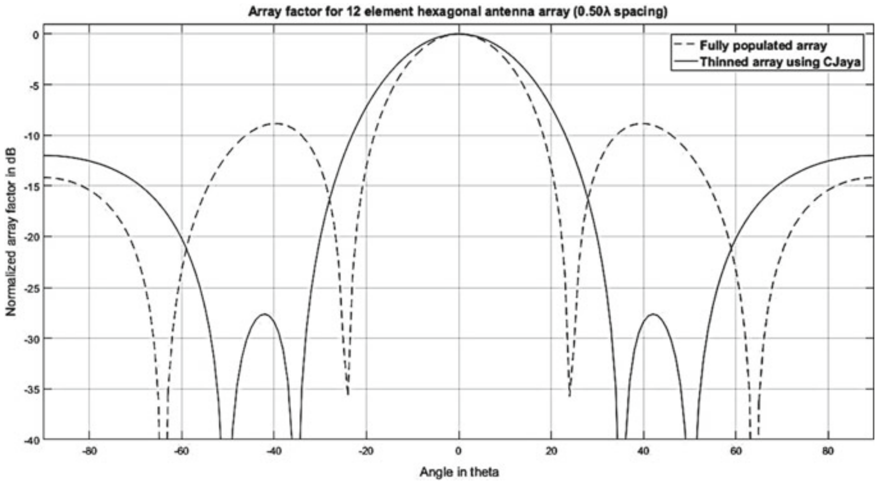

Fig. 5 Normalized array pattern of HA for 0.50λ spacing

Table 4 Hexagonal array's amplitude distribution for 0.50λ

Algorithm	Distribution of ON-OFF elements
Chaotic Jaya	[a]*V*: 1 0 1 1 1 0 [a]MP: 1 0 0 1 0 0
[b]IPSO	*V*: 1 0 0 1 0 1 MP: 0 1 0 0 1 1

[a]*V* vertices *MP* mid-point
[b]The results of elements distribution using IPSO are reproduced from Bera et al. [2]

The inter-element spacing is taken as 0.55λ. Table 5 reports the optimized array patterns obtained by Chaotic Jaya algorithm for reducing SLL for 12-element array and of IPSO [2]. From the table, it is clearly visible that the result of IPSO is −18.03 dB and Chaotic Jaya algorithm achieved a minimum peak SLL of −27.651426 dB which is an enhancement of IPSO results. Figure 6 depicts the radiation pattern achieved by Chaotic Jaya algorithm along with the pattern of a fully populated array. The amplitude distribution is given in Table 6.

Case 3: 0.60λ spacing

Table 5 Results of hexagonal antenna array (0.55λ)

Algorithm	No. of ON elements	% Thinning	Max. SLL (dB)
Chaotic Jaya	6	50	−27.651426
[a]IPSO	8	33.33	−18.03

[a]The results of IPSO are reproduced from Bera et al. [2]

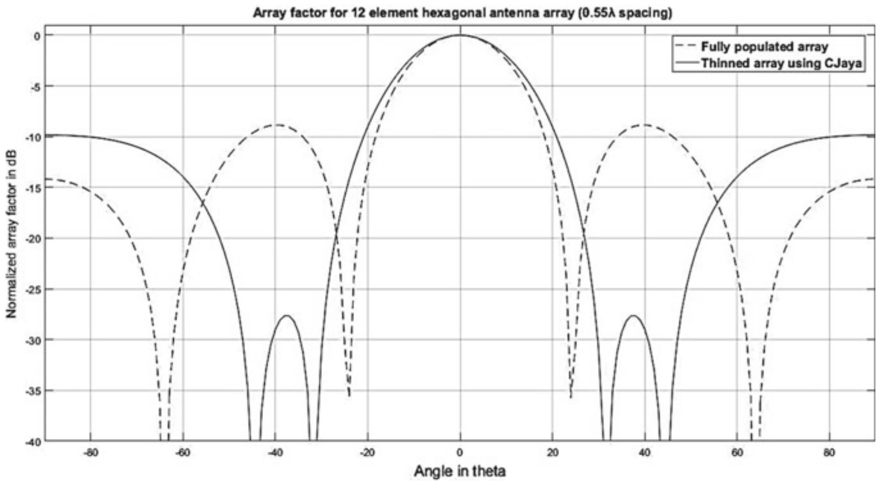

Fig. 6 Normalized array pattern of HA for 0.55λ spacing

Table 6 Hexagonal array's amplitude distribution for 0.55λ

Algorithm	Distribution of ON-OFF elements
Chaotic Jaya	[a]*V*: 1 0 1 1 0 1 [a]MP: 1 0 0 0 0 1
[b]IPSO	*V*: 1 0 0 0 0 0 MP: 1 1 0 0 1 1

[a]*V* vertices *MP* mid-point
[b]The results of elements distribution using IPSO are reproduced from Bera et al. [2]

The inter-element spacing is taken as 0.60λ. The optimized array patterns obtained by Chaotic Jaya algorithm for reducing SLL for 12-element array and of IPSO [2] is tabulated in Table 7. On inspection, it is easily perceivable that Chaotic Jaya algorithm resulted in a minimum peak SLL of −23.866137 dB which is clearly an enhancement over the SLL results of IPSO, i.e., −13.95 dB. The radiation pattern resulting from Chaotic Jaya algorithm align with the pattern of a fully populated array is plotted in Fig. 7. The amplitude distribution is given in Table 8.

Table 7 Results of hexagonal antenna array (0.60λ)

Algorithm	No. of ON elements	% Thinning	Max. SLL (dB)
Chaotic Jaya	6	50	−23.866137
[a]IPSO	9	25	−13.95

[a]The results of IPSO are reproduced from Bera et al. [2]

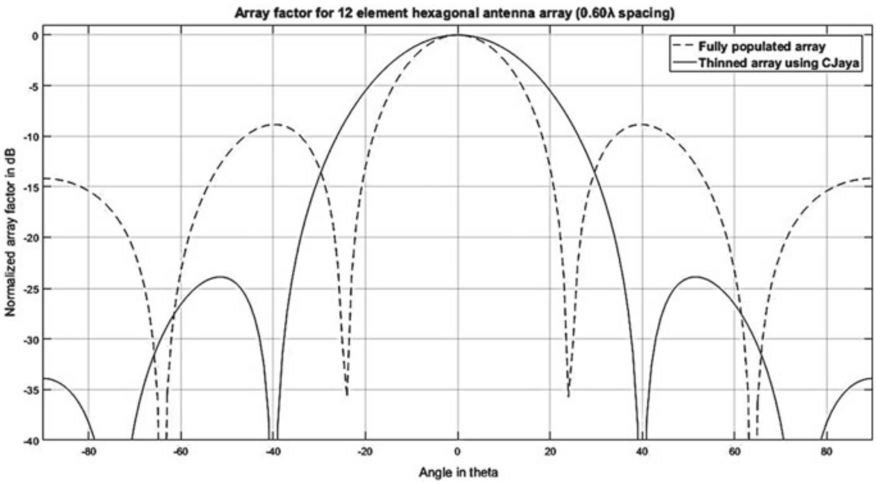

Fig. 7 Normalized array pattern of HA for 0.60λ spacing

Table 8 Hexagonal array's amplitude distribution for 0.60λ

Algorithm	Distribution of ON-OFF elements
Chaotic Jaya	[a]*V*: 1 0 0 1 0 0 [a]MP: 1 1 1 0 0 1
[b]IPSO	*V*: 1 0 1 0 1 0 MP: 1 0 0 0 1 1

[a]*V* vertices *MP* mid-point
[b]The results of elements distribution using IPSO are reproduced from Bera et al. [2]

6 Conclusions

In this work, synthesis of planar antenna array for suppression of SLL is considered. Concentric circular and hexagonal antenna arrays have been synthesized. Four optimization case studies have been formulated and the same are worked out successfully by Chaotic Jaya algorithm. Results' comparison is done between Chaotic Jaya algorithm, FFA and IPSO. It is noted that Chaotic Jaya algorithm accomplished the task by achieving enhanced results than the other above mentioned optimization algorithm with a high convergence speed.

First, the synthesis of two-ring concentric circular array is considered. The number of elements is 35 and 70 on the rings. It is perceived that Chaotic Jaya algorithm's array pattern attains reduced side lobe levels as compared to FFA algorithm's array pattern.

Under the synthesis of 12-element hexagonal antenna array, three case studies with spacings 0.5, 0.55 and 0.60λ are considered. In all the three, increased SLL suppression of SLL is evident from the array pattern of Chaotic Jaya algorithm than the SLL of conventional and IPSO optimized array pattern.

The results show that the Chaotic Jaya algorithm can effectively synthesize a planar antenna array while keeping the desired antenna parameters within acceptable limits. The implementation of Chaotic Jaya algorithm is relatively simple and excludes algorithm-specific parameters. In terms of having good solution search ability and high speed of convergence, the Jaya algorithm is complete in itself. However, with the incorporation of chaotic theory it becomes Chaotic Jaya algorithm and performance is enhanced. Therefore, the Chaotic Jaya algorithm may be further applied for solving optimization problems in the electromagnetic engineering field.

References

1. Basu, B., Mahanti, G.K.: Thinning of concentric two-ring circular antenna array using firefly algorithm. Trans. Comput. Sci. Eng. Electr. Eng. **19**(6), 1802–1809 (2012)
2. Bera, R., Lanjewar, R., Mandal, D., Kar, R., Ghoshal, S.P.: Comparative study of circular and hexagonal antenna array synthesis using Improved Particle Swarm Optimization. Procedia Comput. Sci. **45**, 651–660 (2015)
3. Biswas, S., Bose, D., Das, S., Kundu, S.: Decomposition based evolutionary multi- objective optimization approach to the design of concentric circular antenna arrays. Prog. Electromagnet. Res. **52**, 185–205 (2013)
4. Chattarjee, A., Mahanti, G.K.: Comparative performance of Gravitational Search Algorithm and modified Particle Swarm Optimization algorithm for synthesis of thinned scanned concentric ring array antenna. Prog. Electromagnet. Res. **25**, 331–348 (2010)
5. Elsaidy, S., Dessouky, M., Khamis, S., Albagory, Y.: Concentric circular antenna array synthesis using comprehensive learning particle swarm optimizer. Prog. Electromagnet. Res. **29**, 1–13 (2012)
6. Guney, K., Durmus, A., Basbug, S.: Backtracking search optimization algorithm for synthesis of concentric circular antenna arrays. Int. J. Antennas Propag. (2014)
7. Haupt, R.L.: Thinned arrays using genetic algorithms. IEEE Trans. Antennas Propag. **42**(7), 993–999 (1994)

8. Keizer, W.P.M.N.: Linear array thinning using iterative FFT techniques. IEEE Trans. Antennas Propag. **56**(8), 2757–2760 (2008)
9. Mangoud, M.A.A., Elragal, H.M.: Antenna array pattern synthesis and wide null control using Enhanced Particle Swarm Optimization. Prog. Electromagnet. Res. **17**, 1–14 (2009)
10. Pathak, N., Basu, B., Mahanti, G.K.: Combination of inverse fast Fourier transform and modified particle swarm optimization for synthesis of thinned mutually coupled linear array of parallel half-wave length dipoles antennas. Prog. Electromagnet. Res. **16**, 105–115 (2011)
11. Rao, R.V.: Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Ind. Eng. Comput. **7**, 19–34 (2016)
12. Rao, R.V.: Jaya: An Advanced Optimization Algorithm and Its Engineering Applications. Springer Nature, Switzerland (2018)
13. Singh, U., Rattan, M.: Design of linear and circular antenna arrays using cuckoo optimization algorithm. Prog. Electromagnet. Res. **46**, 1–11 (2014)
14. Jaya Lakshmi, R., Mary Neebha, T.: Synthesis of linear antenna arrays using Jaya, self-adaptive Jaya and Chaotic Jaya algorithms. AEU-Int. J. Electr. Commun. **92**, 54–63 (2018)